Supporting Information

A practical guide to metabolomics software development

NIH Common Fund Metabolomics Consortium Software Standards Working Group (listed alphabetically): Hui-Yin Chang[1,2], Sean M. Colby[3], Xiuxia Du[4], Javier D. Gomez[5], Maximilian J. Helf[6], Katerina Kechris[7], Christine R. Kirkpatrick[8], Shuzhao Li[9], Gary J. Patti[10], Ryan S. Renslow[3,11], Shankar Subramaniam[8,12], Mukesh Verma[13], Jianguo Xia[14], Jamey D. Young[5,15,*]

[1]Department of Pathology, University of Michigan, 1301 Catherine St., Ann Arbor, MI 48109, USA. [2]Department of Biomedical Sciences and Engineering, National Central University, No. 300, Zhongda Rd., Zhongli Dist., Taoyuan City 320, Taiwan. [3]Biological Sciences Division, Pacific Northwest National Laboratory, PO Box 999, MSIN: K8-98, Richland, WA 99352, USA. [4]Department of Bioinformatics & Genomics, University of North Carolina at Charlotte, 9201 University City Boulevard, Charlotte, NC 28223, USA. [5]Department of Chemical and Biomolecular Engineering, Vanderbilt University, PMB 351604, 2301 Vanderbilt Place, Nashville, TN 37235, USA. [6]Boyce Thompson Institute and Department of Chemistry and Chemical Biology, Cornell University, 533 Tower Road, Ithaca, NY 14853, USA. [7]Department of Biostatistics and Informatics, University of Colorado Anschutz Medical Campus, 13001 E. 17th Place B119, Aurora, CO 80045, USA. [8]San Diego Supercomputer Center, University of California San Diego, MC 0505, 9500 Gilman Drive, La Jolla, CA 92093, USA. [9]The Jackson Laboratory for Genomic Medicine, 10 Discovery Drive, Farmington, CT 06032, USA. [10]Department of Chemistry, Department of Medicine, and Siteman Cancer Center, Washington University in St. Louis, CB 1134, One Brookings Dr., St. Louis, MO 63130, USA. [11]Gene and Linda Voiland School of Chemical Engineering and Bioengineering, Washington State University, PO Box 646515, Pullman, WA 99164, USA. [12]Department of Bioengineering, Department of Computer Science and Engineering, Department of Cellular and Molecular Medicine, and Department of Chemistry and Biochemistry, University of California San Diego, 9500 Gilman Drive #0412, La Jolla, CA 92093, USA. [13]Epidemiology and Genomics Research Program, National Cancer Institute, National Institutes of Health, Suite# 4E102, 9609 Medical Center Drive, MSC 9763, Rockville, MD 20850, USA. [14]Faculty of Agricultural and Environmental Sciences, McGill University, 21111 Lakeshore Road, Ste. Anne de Bellevue, Quebec H9X 3V9, Canada. [15]Department of Molecular Physiology and Biophysics, Vanderbilt University, PMB 351604, 2301 Vanderbilt Place, Nashville, TN 37235, USA. *Corresponding author (j.d.young@vanderbilt.edu)

Table of contents

Data formats used in metabolomics S2
1. Vendor formats for raw MS data. S2
2. Open formats for raw MS data. S2
3. Open data formats for MS/MS spectral library matching S3
4. Open data formats in statistical and functional analysis S3
5. Data formats in chemical property prediction S3
6. File formats used for data export and visualization S4
7. Data formats for graphics and reports S4
8. Program-specific data formats S4
9. File size and compression S5
References S5

Data formats used in metabolomics

1. Vendor formats for raw MS data.

Data preprocessing software takes raw instrument data as input and extracts qualitative and quantitative information from it. Instrument manufacturers use different proprietary formats for raw MS data. **Table S1** lists vendor-specific proprietary MS data formats.

Proprietary Format	MS manufacturer
.raw	Thermo Fisher Scientific
.d	Agilent
MassLynx .raw	Waters
.wiff	SCIEX
.peg	LECO
Compass .d	Bruker
.lcd	Shimadzu

Table S1. Commonly used proprietary MS data formats.

2. Open formats for raw MS data.

MS instrument manufacturers provide their customers with software tools to read and preprocess raw MS data generated from their instruments. Despite the utility of these vendor software packages, numerous other software tools have been developed to provide vendor-agnostic visualization and processing capabilities, many of which are not available in vendor software tools. As MS technologies continue to advance, there is a growing need to develop vendor-agnostic tools for processing increasingly complex MS data sets.

Development of vendor-agnostic software packages requires that proprietary MS data formats be converted to open data formats. MS data in open formats can be read using freely available software tools or modules, which greatly facilitates the development of vendor-agnostic solutions. Standard open data formats that are widely used include mzXML, mzML, and NetCDF. The mzXML format was developed at the Institute of Systems Biology (ISB), while mzData was developed by the Proteomics Standards Initiative (PSI) during 2003−2005[1]. The mzML format was developed at ISB to unify mzXML and mzData by preserving the strengths and avoiding the weaknesses of each format. Since mzML version 1.1.0 was released in June 2009, it has been stable with wide adoption by software developers, and is currently the *de facto* standard MS file format. Like mzXML, the mzML format encodes information in Extensible Markup Language (XML) format and is designed to contain both spectra and metadata about these spectra from a single MS run. The spectra themselves can be saved in binary format by using the zlib compression, but the metadata and XML tree structure are still in legible text format. As a result, mzML does result in larger file sizes in comparison with vendors’ proprietary formats and can suffer from slow read/write speeds. Some groups have transitioned to performance-oriented open data formats (e.g., HDF5) for applications where mzML file sizes become unwieldy.

The NetCDF (Network Common Data Form) format is another open data format that is widely used and well supported by software tools for MS data preprocessing[2]. It was created to support the creation, access, and sharing of array-oriented scientific data. It is a binary format and therefore does not result in a drastic increase in file size when converted from a vendor format. In addition, it contains some metadata about the instrument or spectra. Because of these strengths, the NetCDF format is used frequently to store GC-MS spectra. However, NetCDF is not equipped with the capability to encode tandem mass spectra. As a result, it is rarely used for storing LC-MS spectra even though it can handle level-1 LC-MS spectra.

Conversion of data from proprietary to open data formats can be accomplished using either vendor software packages that are equipped with such capability or ProteoWizard, which is a widely used free conversion tool. The ProteoWizard module that is specifically developed for converting data from vendor to open data formats is msConvert and works only on computers that run the Windows operating system. In addition, ProteoWizard contains a module called seeMS that can directly read and display MS data that are in the original proprietary data formats.

In addition to seeMS, there are many programming language-specific tools that can read open data formats. These tools include pymzML (Python), jmzML (Java), openMS (C++), mzR (R/Bioconductor), netCDF4-python, netCDF4-Java, netCDF C++, among others.

3. Open data formats for MS/MS spectral library matching

Library matching for identifying/annotating mass spectra usually takes input files in .msp or .mgf format. The files hold information on individual spectra with a variable set of metadata. They are often used for spectral libraries of identified compounds, but their flexible and easy-to-parse format can be used for data exchange in other settings as well.

4. Open data formats in statistical and functional analysis

Results and input from statistical analyses and pathway analysis can often be contained in delimited text files, for instance as tab-separated values (.tsv or .txt) or comma-separated values (.csv). The advantage of these formats is their ease of access for users of all skill levels, since they can be viewed and further analyzed with a wide range of other software tools, including text editors and Microsoft Excel. Delimited text files work best for data that can be displayed in a table format with one data value per cell (i.e., "flat" data, as opposed to data that contains more complex data structures for each entry). The default delimited text file contains one header line with column names, followed by lines of data for each column, optionally including row names. Each column should contain data of one type (e.g., only numeric or only boolean values) to facilitate parsing during file import. Some metabolomics software tools have specific layout rules that can accommodate metadata in delimited text files expected by the importing program. For instance, text files for Meta-Analysis in MetaboAnalyst require a second row of headers specifying sample grouping information[3].

The mzTab-M format is a metabolomics-specific, .tsv-based data format that was developed in collaboration with the Metabolomics Standards Initiative consortium[4]. It is an extension of the proteomics-centered mzTab format with some metabolomics-related specifications. The format is human-readable when opened in text editors or Excel, but still allows saving of metadata along with quantification information and evidence of molecular identity, using multiple Table sections within the document that are linked through identifier numbers between entries. Some popular analysis tools already support mzTab-M, and application programming interfaces (APIs) exist for Java, Python, and R.

For more complex data it may be useful to use XML-based file formats that allow hierarchical organization of data using a tree-like structure. Each node can have an arbitrary number of attributes, which can contain relevant metadata. XML-based formats are often used to store large amounts of raw data, for example with nodes holding spectra in a space-saving binary data format. Examples include the widely used mzML and mzXML formats for MS data files or the graphML format that is used to save network graphs, including results from MS/MS-based molecular networking. While they are human-readable in principle, their structure is more accessible through specialized APIs.

5. Data formats in chemical property prediction

Most chemical property prediction software requires a representation of molecular structure. There are numerous molecular structure file formats available now, and it is relatively easy to convert between these formats using tools such as OpenBabel and RDKit. However, frequently used formats include .xyz, .mol2, .sdf, and .pdb. XYZ files are the most common for quantum chemical software, but they only store nuclei coordinates and not bond information. Note that converting to XYZ from other formats is reliable, but conversion from XYZ to other formats that hold bond information is unpredictable as it relies on assumptions about nuclei distances to guess bond types, which can frequently fail. It is recommended to use tools such as JANPA or NBO in these cases so that the electronic structure can be used to predict bond types. To store large numbers of molecular structures we suggest the use of SMILES (Simplified Molecular-Input Line-Entry System), a chemical structure line notation. InChI (IUPAC International Chemical Identifier) is another textual identifier, but is not as robust or facile for cheminformatics software.

6. File formats used for data export and visualization

User interaction with software tools ultimately results in the extraction of information from the raw data. This can be as simple as text-based feedback, a plot, or a report file. In many cases, the results will be used in downstream data analysis, either manually or using other software. In these cases, the program outputs need to be packaged into machine-readable file formats. It is good practice to keep the format as simple and universally accessible as possible to ensure that users can view, process, save and report the analysis results. However, some information can best be represented as visual outputs or saved in software-specific files (**Table S2**).

Format	Examples	Use case
Plain text	.txt, .tsv, .csv, mzTab-M	Human readable, flat format
XML-based	mzXML, mzML, graphML	Hierarchical structure, mixing binary and text data
Graphics and reports	.pdf, .png, .jpeg, HTML	Visual representation for humans
Program-specific	.mtd (MSDIAL), .mzmine	Machine-readable information for a specific software

Table S2. Commonly used formats for data export.

7. Data formats for graphics and reports

When graphical user interfaces are included in the program, meaningful plots provide quick feedback to users so they can adjust their analysis settings as needed. Ideally, these plots should be exportable as a vector graphic to make them immediately suitable for use in publications. Such implementations minimize the burden on the user to visualize their data and help to standardize program outputs across the community. Additional report formats can be attractive, for instance, multi-page .pdf files with plots and tables that are human-readable. Optimizing the layout of these files and writing unit tests for them can be challenging and is most valuable if the automated output replaces work that would otherwise be repetitive and time-intensive. HTML-based output formats, or even a web interface, can be useful to include interactive content and graphics in program outputs. However, even the most sophisticated of graphical outputs should not replace a way to export the data in a machine-readable format, because it allows users to explore visualization or statistical analysis of the results in other software (e.g., Cytoscape or R).

8. Program-specific data formats

Results that are exportable into files that can be saved and reloaded make it possible to continue data analysis at a later point in time or on a different computer. This is particularly relevant if the analysis is time-consuming or difficult to repeat. Temporary storage of intermediate results is often achieved by program-specific formats that can hold all relevant data and metadata, often structured in a way that minimizes time for saving and reloading. For instance, any R object can be serialized into a .rds file with a base R function, Python objects can be saved as Pickle files (.pkl) as the standard format, or MATLAB files can be stored as binary .mat files. The drawback of this approach is that the program outputs cannot be easily re-opened by other software, and it is unlikely that other software tools would implement importer functions for these specific formats.

9. File size and compression

Regardless of the output format, there are some file size and compression considerations that must be taken into account. Depending on the type of analysis, the output file size can be an important factor to facilitate storage, sharing and reloading of data. For small outputs (e.g., program settings used for analysis) optimizing for file size is not necessary. However, when large amounts of raw data are included in the output, the benefits of minimizing file size can outweigh the additional compute time and programming efforts that are typically required for compression. The most obvious way to reduce file sizes is to avoid redundant information in the file. For program-specific outputs, this would involve not including information that can be derived from the rest of the data by the program, especially if recalculations are fast. Text formats are generally larger than binary formats but can benefit greatly from compression (e.g., as a .zip file).

References

(1) Ranganathan, S.; Gribskov, M.; Nakai, K.; Schönbach, C. *Encyclopedia of Bioinformatics and Computational Biology*; 2018. https://doi.org/10.1016/c2016-1-00174-8.

(2) Brown, S. A.; Folk, M.; Goucher, G.; Rew, R.; Dubois, P. F. Software for Portable Scientific Data Management. *Comput. Phys.* **1993**. https://doi.org/10.1063/1.4823180.

(3) Chong, J.; Wishart, D. S.; Xia, J. Using MetaboAnalyst 4.0 for Comprehensive and Integrative Metabolomics Data Analysis. *Curr. Protoc. Bioinforma.* **2019**. https://doi.org/10.1002/cpbi.86.

(4) Hoffmann, N. et al. MzTab-M: A Data Standard for Sharing Quantitative Results in Mass Spectrometry Metabolomics. *Anal. Chem.* **2019**. https://doi.org/10.1021/acs.analchem.8b04310.